Discussion Notes

Here is some background information, based on the widely known essay by Eric Raymond "The Magic Cauldron."[1] and a report from the Sustainable Business Models Team to the Brokering Governance Working Group of the Research Data Alliance (RDA) [2]

According to Raymond, there are the following business models

- Loss-leader/market positioner: use open source software to create or maintain a market position for proprietary software (e.g., an open source client creates a market for a proprietary server)
- Widget frosting: publish open source drivers for proprietary hardware, both for peer review benefits and also to allow operating system vendors/maintainers to adapt the driver to future changes in system interfaces
- Consulting, also known as "give away the recipe, open a restaurant:" use expertise in an open source product to drive revenue for packaging and/or consulting services (e.g OpenFOAM)
- Accessorizing: sell books or other accessories to open source products (e.g., O'Reilly publishers)
- Free the future, sell the present: sell a proprietary product with a license that guarantees open source release after a certain time, in order to guarantee future maintainability to prospective customers (e.g., Alladdin GhostScript)
- Free the software, sell the brand: charge for the branded, trademarked, tested, and certified version of an open source product (e.g., RedHat)

Raymond's essay also pointed out that in most cases, open source software does not have as much revenue available to fund its development as is provided by license fees of proprietary software. For this reason, proprietary software often leads the development of end-user software, while open source provides low-end users with an inexpensive alternative, and also provides an open and flexible platform for those who need custom modifications. Proprietary products must therefore keep innovating to push the frontier forward, as its open source competitors catch up behind them.

[1] E. Raymond, "The Magic Cauldron," The Cathedral and the Bazaar (Sebastopol, CA: O'Reilly, 1999), http://catb.org/~esr/writings/magic-cauldron/
[2] https://www.rd-alliance.org/group/brokering-governance-wg/outcomes/sustainable-business-models-brokering-middleware-support

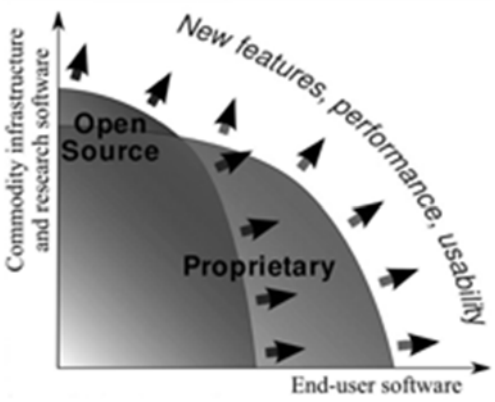

Figure 1: Eric Raymond's concept of expansion of open source and proprietary software capabilities

The RDA report discusses five classes of business models. For each, the strength and weaknesses in the context of long-term sustainability is discussed. For their type of software (brokering middleware) they find that a hybridized model incorporating aspects of three different business models over the lifespan, i.e. federally funded data facility guardianship in the establishment stage replaced or supported by a Consortium model and/or Software-as-a-Service as the software matures, will likely provide the strongest model for sustainment.

Figure 2: Required attributes for software sustainability adapted from Ref 2 and the Software Sustainability Institute's definitions for software sustainability (http://www.software.ac.uk/)

Sustainable Business Models

Software Business Models

Software services for research provide intangible goods and services, and frequently employ business models different from those utilized by companies providing goods and services focused on financial, physical or human sectors. As they are currently configured, those entities responsible for software serve or could serve four classic business archetype roles.

● A creator that transforms ideas into a product

● A distributor of software

● A lessor that provides the rights to use the software

An underlying premise of the archetypes with respect to software is that the provenance and intellectual property (IP) are established.

Software businesses may use a hybrid approach serving the role of more than one archetype function since they are acting as both the inventor and the IP lessor. Software companies can differentiate their business model by offering software as a product or, software as a service (SaaS), or a combination of both. As K.M Popp (2015) notes, SaaS means the software vendor does not deliver the software, but the customer gets both access to the software and usage rights ..... The software vendor carries the cost of software support, maintenance, and operation."

Revenue Models

The software industry has developed a number of business models to generate revenue to support all or some of the attributes of sustainment outlined above. SaaS is considered in this review of viable revenue models, as well as non-commercial and hybrid funding models. The revenue models explored are:

- Government funding through assistance awards and contracts
- Government funding through federal data facility guardianship
- SaaS (including tiered pricing);
- Information and Ad sales
- Corporate Support (including Foundation Support) and Product /Service Sales, (foundation support)
- Consortium model

Each of these revenue options are discussed in the following context and may have variants depending on the target market:

1) Description of the business model with examples of software currently supported under that model
2) Mapping Revenue Models onto Sustainment Attributes
3) Characteristics of the business model that would benefit the sustainability of software
4) Challenges in the application of the model to support software for the research and education communities

Government Funding Through Assistance Awards and Contracts

Model Description

For decades, governments have provided funding to support the development and sustainment of software. In the U.S., this support takes the form of contracts, grants, or cooperative agreements. Contracts provide services to the government, such as the development and maintenance of software and hardware used to ensure safe air travel. Grants are assistance awards that, in the context of the interests of the Research Data Alliance, are often used to "assist" university and non-profits in the pursuit of a research problem. The supporting government entities generally have little involvement in guiding the research activity that is funded by grants. Cooperative agreements are also assistance awards, which are generally reserved for use in the support of large and complex undertakings. In these cases, the government has substantial involvement in shaping the course of the research endeavor in cooperation with the investigators.

Aside from developing software in the research context of computer science such as compiler languages, most software development is a means-to-an-end to facilitate an inquiry, such as prove or disprove a research hypothesis, to gain insight into relationships within data sets, facilitating connections among essential but disparate data sets, or to simulate a physical phenomenon. Globus, NetCDF, and the Community Climate Model are examples of software that have been developed with government support and continue to be refreshed and modified with U.S. Federal dollars. The longevity of support for these software packages is measured in decades.

Occasionally governments will invest more strategically to support the national advancement of science research and development. In Australia, the Commonwealth Government provided 5-10 years funding for the National Collaborative Research Infrastructure (NCRIS) and Super Science 11

Initiatives, which have supported important national research data initiatives such as the National Computational Infrastructure, Terrestrial Ecosystem eResearch Network, Australian Urban Research Infrastructure Network and the Integrated Marine Observing System. More importantly, this funding has significantly enabled Australian researchers to participate in eResearch by supporting the development of IT infrastructure, software, tools and services for open publication, discovery, sharing and use of Australia's research data outputs through the Australian National Data Service (ANDS) and National eResearch Collaboration, Tools and Resources (NeCTAR). The US National Science Foundation is funding the EarthCube Program, which provides assistance grants for the development of Cyberinfrastructure to advance the geosciences. The program is expected to last a minimum of 10 years and has a strong focus on improving interoperability and data sharing across geoscience domains.

Mapping Revenue Models onto Sustainment Attributes

Funding – To obtain support from the government, a proposal is required. The home institution of the principal investigator bears part or all of the cost of preparing the proposal through its overhead charges (although overhead covers more than proposal development).

Government initiated strategic investments, like the NCRIS and Super Science initiatives in Australia, can be of longer duration than research grants and cooperative research agreements, but are nonetheless limited in their duration.

Users – With respect to software developed as part of a research grant, the concept of market development is interpreted to mean understanding which research communities might find the software useful in the pursuit of their objectives. This is an important observation that could have an impact on sustainability. The design of the software could be influenced by the applicability of the software to other venues. This takes forethought on the part of the principle investigator either at the outset of the project, for example in the proposal, and/or realization that there is greater need for the functionality of the software than originally anticipated. Government funding does not necessarily encourage or discourage the principle investigators from engaging in market development [but some funding agencies, such as NASA, encourage and may require that software developed under the grant be made available to the community for reuse]

Communities – An important consideration by the government with respect to sustained support is the extent of community engagement. The engagement is not only concerned with the number of people using the software, indeed might depend on its availability, but also the influence that community of users has over the future directions of software capabilities and refinements. Often the government oversight of a project will encourage community engagement and might even support additional funding for outreach and engagement activities that might come in the form of funding for a workshop or seminar series. ANDS and NeCTAR have played a significant role in building enabling Australian researchers in the eResearch space, including building and maintaining the communities that will implement and shape data practices, standards, tools and infrastructure into the future.

Human Resources –Software developed under a research grant or cooperative agreement usually receives financial support for the duration of the award, which can range from one to five years. Support can come incrementally on an annual basis or be provided in entirety at the beginning of the award. The incremental support is subject to adequate progress in research activities. Both grants and cooperative agreements allow modification of the original proposal, subject to approval by the funding agency. Most often the award is made to the institution which employs the researcher. Often this institution provides tangible and intangible contributions. Under these circumstances software developed under a research project 1) has the revenue stream for the work virtually assured for a limited period of time, 2) can be flexible in design and execution, 3) has a natural partner in the institutional home that often provides benefits useful to the project, and 4) has the potential to contribute to the educational mandate of the institution.

Software Engineering – The application of the best practices in software engineering is an essential element for sustainability. Although less rigor might allow short term objectives to be achieved, continued support might hinge on the ability of the software to be easily modified to meet researcher's changing needs and/or emergence of new data and technologies. Therefore investing in good software engineering is an important use of the revenue obtained from the Government.

However, government funding of a project, particularly in a university setting, does not assure that good software engineering practices will be used throughout the project. The practices of educational institutions to have students and research assistants write software usually means that the engineering practices are either unknown or not followed by the software authors. If the software is not meant for a broad audience, rigorous development practices are less likely to be followed.

Product Management – Under government funding, product management is usually the responsibility of the institution that received the funding and the principle investigator employed by the institution. The government often encourages the sharing at minimum cost of software developed under a grant or cooperative agreement. However, modified versions of software might be offered as a commercial product. In general, the government usually allows the Intellectual Property rights to reside with the institution where the software was developed but this may require the government to have royalty free access to the software. The on-going review process employed by government agencies encourages principle investigators to demonstrate good product and project management.

Advantages

Advances the conceptualization, prototyping, and initial testing of the software for such attributes as functionality and robustness, since there are no other sustainment attributes of the software that need to be considered during the initial development and demonstration phase.

There are some instances of longer-term support of software that the funding agencies deem necessary for significant community functions such as weather prediction and for these, concern must be paid to each of the sustainability attributes, but the assured funding stream allows the project management to navigate the allocation of resources to addressing each of the attributes.

Challenges

Funding term limits do not create sustainment. There is a significant risk that government funding is not a predictable or reliable long-term solution for sustainability.

An all too familiar scenario is the demise of software developed as part of a research grant after the research grant expires. Even though grants and cooperative agreements have limited duration (maximum of five years), perpetuation of activities under previous awards is possible through follow-on support. Long-term support of software as an essential element to achieve scientific research objectives is possible under certain conditions determined by the funding agency's priorities, policies and practices. The Globus, NetCDF, and Community Climate Model examples cited above share certain features: 1) They provided very useful cyberinfrastructure to advance research and education activities; 2) Their sustainment has been part of a larger set of activities, e.g. NetCDF is supported through the Unidata award and the Community Climate Model is maintained as part of the award to support the National Center for Atmospheric Research; and 3) the funding agencies recognize the need to invest in infrastructure (physical and cyber) to sustain the health of scientific enterprise. The challenge is that these examples are the exception and not the rule and Globus is now exploring traditional mechanisms of sustainment outside of government funding to complement its government support. This is particularly true when the software, such as middleware, is not as visible to users in successful operations. Furthermore, Government initiated strategic investments like the NCRIS and Super Science initiatives are often politically motivated and at the mercy of changes in Government leadership and party politics. The size, complexity and altruistic nature of the programs (e.g. ANDs and NeCTAR) established under such funding arrangements, makes it extremely difficult for these programs to continue under other funding arrangements after federal government funding is withdrawn.

As noted earlier, government support for software development does not ensure that good software engineering practices are employed, particularly if the software is developed as a means to an end when, for example, addressing research questions. It is not uncommon for software used in a project to be developed by graduate students with little if any software engineering experience and have almost no documentation. To be sustainable, software needs to be well engineered and well documented for its present purposes as well as future uses. In this case, future uses may mean crossing disciplinary boundaries to reach a broader user base. Grant funded software may not address this. The required software engineering must be explicitly planned for, but might not receive funding from the agency if the focus of the project is scientific research. The development should also address reliability and ease of operation. When software is created for one purpose and used by its developer, ease of operation by others or an outside organization is not generally a concern of the developer.

Using grant support there can be less need to identify to funding agents how resources were allocated toward the various attributes that are need to develop, maintain, and sustain the software nor to seriously consider the long-term implications of market development and product management including revenue models.

Government Funding through Federal Data Facility Guardianship

Model Description

Federal applied science and data agencies such as the National Oceanic and Atmospheric Administration (NOAA), National Aeronautics and Space Administration (NASA), Australia's Bureau of Meteorology, the U.S. and British Geological Surveys, and Geoscience Australia develop software and systems to manage, analyze and provide data and information to government, industry, education, research and the public, so as to maximize the economic and social benefits the agency delivers nationally. Federally funded research data facilities such as those funded by the National Science Foundation and NASA, serve specific communities within scientific domains. Their mission is driven by the needs of the community they serve. Government data facilities supported by the NSF are typically governed by a Board of Directors who determine the budgetary priorities of the organization based on requirements put forth by the community and the data facility management.

In the guardianship funding model, the data facility may adopt and/or develop software to serve their community. The costs of maintaining the software would be incorporated in the operations and maintenance costs of the data facility. Success of this model requires a strong advocacy from the community as well as within the data facility. Sustainment is assessed by the Board of Directors based on continuing assessment of the value to the community.

Federal agencies will occasionally invest in software that delivers benefit to all government stakeholders, for example, in support of the open data agenda as a means of stimulating economic development.

Similar modalities are seen in Europe both at the national level and at the European Commission. Europe supports developments for e-infrastructure and complementary activities in research and application of data. The research and development grants do not address operational support.

Mapping Revenue Models onto Sustainment Attributes

Funding – Funding is provided by the funding agency of the data facility and budgeted as part of operations and maintenance. Agencies manage their budgeted allocations to deliver sustainable long-term objectives whilst delivering projects and the Government of the day's policies and strategic priorities.

Users – Not actively undertaken since support is provided by the requirements of the user community. However, occasionally governments will solicit ideas for software development that would address unmet needs of current software and/or the development of software that expands or complements existing cyberinfrastructure.

Communities– US Federal research data facilities are very well connected to their primary user communities and are dependent upon these communities for guidance (Board of Directors, Steering Committees, strategic planning). These facilities cannot continue to exist without the strong support and use by their communities. Hybrid approaches have been tried in the US, for example, software development directly by the agency or through contract and/or grants is provided (source code and all documentation) directly to community groups to maintain and modify such that it best serves the needs of the community over time.

Human Resources – Software and systems development and maintenance is planned, budgeted and implemented on an annual basis to meet the prioritized needs and objectives of the data facility and community it serves. Externally developed software may also be incorporated into the operations of the data facility.

Software Engineering – Software development and/or maintenance is provided internally by an expert/advocate within the data facility, such as the central ICT unit, or more frequently is contracted to private companies.

Product management – Initial product planning and design may be undertaken through the support of a grant but long-term management is sustained through the data facility. This activity is guided by community driven requirements and implemented by the governing body of the data facility. The data facility manages the licensing, version control and distribution of the software.

Advantages

- Federally funded data facilities tend to be long-lived, secure, well-maintained and are appropriately resourced for as long as the provision of data remains a government priority. In the case of data repositories, it is the organization's mission to manage and maintain their data and associated data discovery and access tools.

- Federally funded data facilities are well-known, visible, accessible and respected by the research community. It is assumed that they represent the authoritative source of data and information.

Challenges

- Finding a home for software within a data facility that serves a specific community will be challenging if the software should serve numerous communities at once, and therefore may not

be highly valued by a single community willing to take on the responsibility and expense of maintenance.

- Federally funded data facilities are typically long-lived but not necessarily permanent. Many operate on a 5-year funding review cycle. For example, Australian Government Departments and their Portfolio Agencies are not guaranteed stable entities. Restructures can result in the loss of entire functions and their associated ICT infrastructure, systems and services.
- These facilities are under constant budgetary constraints and the value of software must remain high to justify continued support. Even within a relatively long term, stable agency like Geoscience Australia, budget allocations and strategic priorities are volatile and can result in the removal of resources supporting software, systems and services if other activities are considered to be a higher priority for the fiscal year. This tends to result in limited or no software development or system maintenance after the launch of a system or tool, and the agency's ICT unit typically maintains an increasing number of incomplete, outdated and often duplicated legacy systems and web applications.
- The software requires a strong advocate and expertise within the data facility to be maintained
- Facilities may wish to develop software in house (provided more control and in-depth knowledge of the code)

Software as a Service (SaaS) Business Model

Model Description:

A common model for software and services sustainability in the private sector is called "Software as a Service" or SaaS. While there are multiple definitions of SaaS, a common one that is useful for the present discussions is:

Software as a Service (SaaS) - software that is owned, delivered and managed by one or more providers and is available remotely. The provider offers access to the software that is consumed in a one-to-many model at any time on a pay-for-use or subscription basis.

While the definition focuses on "pay-for-use", there are almost as many variants as there are markets. This will be discussed below. However, there is a general set of characteristics for SaaS that are crosscutting:

- Hosted
- On demand
- Integrated (operates on a platform)
- Subscription or other form of relation
- Multi–tenant (simultaneous use)

- Supports Network effect i.e. builds more rapidly, leveraging marginal benefit and tipping point evolution (viral adoption)

These characteristics become particularly interesting as research and applications move to using Big Data, and cross-discipline modeling and processing emerge as essential needs to address global issues. From a business perspective, the cloud paradigm and implementation provides a rich environment for customer interactions and support. This allows for more dynamical support and removes geographic boundaries, which could inhibit more traditional hardware businesses.

SaaS has found wide applicability in a range of markets. Examples of SaaS implementations include WordPress (a website resource), Survey Monkey, Smartsheet, Customer Relations Management, GLOBUS, iPlant, GitHub, ArcGIS Online, FME Server and many others. Generally, these provide direct and visible utility to end-users and are proving themselves to be an economical replacement for in-house software and infrastructure in government science agencies and data repositories. There is a challenge in comparing pricing and the markets mentioned above and the market environment for middleware because the level of visibility is different, and in fact, the customer base may be different. For example, if we consider the need for interdisciplinary research, there may not be a broad base of existing resource infrastructures that reach across research domains.

This challenge does not preclude a service-based approach and related pricing. For example, the pricing can be based on incentives to procure higher-level capabilities and services. The initial service offering can be a free service that engages the community. Once engaged, options for more capabilities are provided (while still retaining the free baseline option) and the user or institution becomes part of a paying base for a sustainable business. The pattern of free offering followed by paid enhanced capability is known as “Freemium”. Examples of various models are given in Table 1 (Pricing Models).

Table 1: Pricing Models (includes Freemium pattern)	
Capacity-based Model	Customer cost scales as capacity, usage, or number of users reach certain thresholds.
Feature-based Model	Customers scales according to the number of key features available
Time-based Model	Subscription with a fixed (extendable) time
Use-case Model	Fees are based on specified categories of customers (non-commercial use, educational, non-profit, etc)
Quality of Service Model	Customers get a certain level of response time or service level

A typical example would be the product smartsheet, an 'intuitive online project management tool enabling teams to increase productivity using cloud, collaboration, & mobile technologies" offered by Smartsheet. This online version of Microsoft Project that enhances collaboration capabilities. The customer has a free trial period and then has the option to chose from of a basic ($14/month), team ($39/month) or enterprise package for continuation. This is a combination of time-based and feature-based pricing models.

Mapping Revenue Models onto Sustainment Attributes:

Funding – The source of funding for SaaS business models includes complementary mix of contributions mentioned above. In a reasonably large number of cases, the approach to funding is a "freemium" model; the fees for enhanced services are not generally high and thus the transition from free service to enhanced services has a low barrier. A SaaS organization benefits from the scale of the number of transactions or subscriptions.

Users – Unlike services to a broad range of end users, the business model for middleware will need to target large institutions and businesses that will benefit from the interoperability offered. For the academic and research community, the market could be institutional clients and data/information repositories. The market development must follow the traditional cultural pattern of the target market and thus will need to be adapted across a range of markets to be effective. For example, for institutional data repositories, the issue of open source software arises frequently and thus the business model must address how to provide enough unique benefits from the service that customers will not simple download the software and apply their own software engineering.

Communities – For customers from the research community, community engagement through presentations at conferences, working sessions, demonstrations and other direct interactions are part of the culture. Community advocacy can support market development.

Human Resources – The SaaS model, particularly for open source modalities, can adopt support and contribution from user and supplier communities. This can come in the form of software recommendations, feedback on usability and adaptation to the cultural norms in target research audiences. There are specific examples of such contributions that have significant impact on the software application effectiveness. There is a model with community software contributions, but it is not a fully open environment. Other models, such as WordPress, have a fully open software core and actively encourage community engagement. Like the evolution of crowd sourcing, community contributions are done on a voluntary basis. For the research community, the mixed model of community contributions and a SaaS modality can be complementary and provide a broad customer engagement for sustainability.

Software Engineering – The role of software engineering is to develop and maintain an operational capability of the software. In the open community model, software can be developed through community contributions and in widely used systems, there can be community inputs on quality. However, software engineering is usually a focused effort and may need more structure than occurs in a virtual and distributed community effort. This should be addressed as part of the program management activities.

Product management –"Traditional" SaaS implementations generally use conventional product management approaches. Typically changes are prioritized and then an agile process may be used to facilitate fast development. Documentation for users is usually web based and accessible through the internet. Chat rooms are a common means of offering answers to users' questions. The application of such management processes to open source software has occurred and is done on an application-to-application basis.

Advantages:

- SAAS models cover a wide range of both open source and proprietary software capabilities
- Flexibility of pricing from very modest costs for supporting an open source community to more structured implementations
- Shorter development cycles when software can be pushed out to the cloud with smaller batch sizes, faster feedback and high overall quality

Challenges:

- New capabilities may not be relevant to a subset of users and they must adapt to the change
- The research community may be a low volume customer. There will be issues to be addressed on how to get higher volumes and flow through
- Need for a set of software interfaces or application interfaces to provide a broad interoperability capability. Building such a base can be time and resource intensive as it moves beyond traditional standards

Information and Ad Sales

Model Description

Online information and ad sales are forms of marketing that use the Internet to deliver marketing messages to consumers and to provide contacts for targeted marketing - individuals that may be interested in purchasing products and services. Online advertising includes web banner advertising, search engine marketing, and other forms. Income may be generated by pay per click, search and web analytics, cost per action, revenue sharing, email and referral marketing, and related mechanisms. Information may be collected on data usage, and may be of interest to either data providers (e.g. usage stats) or users (popular tools or data), potentially providing an additional revenue stream.

Information and Ad Sales is the business model used to support well known software products such as:

- Google
- Facebook
- YouTube
- annotation/comment service (e.g. Pinterest)

Mapping Revenue Models onto Sustainment Attributes

Funding – Funding is provided by the business or industry that benefits from information collected on users (e.g. email addresses) or exposure to corporate ads that result in traffic to the business or industry and subsequent sales. Business contracts must be sought and created and the business or industry that is seeking information or placement of ads.

Users – Considerable effort may be required to identify businesses or industries that desire the information that can be obtained and to identify appropriate advertisers.

Communities – Caution must be exercised in selling user information to external entities or integrating ads into services as such activities may be viewed as violating personal privacy or creating negative associations with businesses and industries.

Human Resources – Additional human resources may be required to capture and package data and information desired by external business and industry interests.

Software Engineering – Software development or contracts with third parties (e.g., Google analytics, web developers) may be necessary to capture/package information and support advertisements.

Product Management – Services such as provision of data and information and supporting advertisements require significant relationship management,

Advantages

● automated mechanism and requires little human intervention

● scales easily as user numbers increase

● may include a user model where users pay to not see ads

● there is potential for annotation service provided by this model to become part of institutional quality assessment process

● this model can contribute to a business diversification strategy for resilience and growth

Challenges

● requires sales to advertisers and/or people that wish to use the service -- requires a sales team and an identified large consumer base

● requires good planning of communications and branding

● ads may disenfranchise user base

● sales of information is a major privacy issue and may disenfranchise user base

● questions about who owns the usage and annotation data

Corporate Support and Product /Service Sales

Model Description

Corporate Support –

In the context of support for a service provided through funds contributed by corporate entities there two primary models: sponsorship and membership, where membership may be broken down further into direct and indirect models. The sponsorship model is one where corporate entities provide general or specific funding for the system as a whole or some subset of the system for which their sponsorship (and acknowledgement) is seen as being aligned with their mission or business model. Often the sponsorship model does not include substantial participation in the governance of the system.

The membership model may provide more direct participation in system governance, depending upon the specifics of the model. One type of membership model is one in which corporate entities "join" an organization that is responsible for the development and maintenance of the software system, potentially at different membership levels that entail different rights and privileges. This "direct" membership model may have substantial member participation in system governance as a benefit of membership. A second type of membership model is an indirect one in which members of corporate entities (e.g. professional organizations, research organizations, universities) are provided access to services of the platform that they would otherwise need to pay for individually or not have access to at all if it weren't for their organizational affiliation. In this indirect membership model the organizations to which the members belong would provide support for a system capability.

Product/Service Sales –

A product or service sales model is another approach for obtaining funds needed to sustain a software. In this instance, the business model includes an outright sale of software for customer platforms with the customer having ownership of the software and then service to support operations on the customer's platform is provided by the vendor. The service could include a schedule of software updates or responding to special customer needs. SaaS, on the other hand, is where the vendor retains ownership and operates the software in a "cloud" or vendor platform. The service modality is quite different with automatic updates of software capabilities, etc. In the fee for service variant users would pay for access to some or all of the services provided, with the amount paid potentially falling into access tiers associated with levels of service or differential access to service types. In this respect there are some similarities between the fee for service mode and the Software as a Service model described above, with the potential differentiating factor being that the financial contributions in this case are primarily coming from product users while the SaaS model has a mixture of end user revenue and organizational users.

Mapping Revenue Models onto Sustainment Attributes

Funding– In the corporate support funding model substantial outreach and negotiation is required to first establish a relationship with potential corporate sponsors/members, then demonstrate the value of the system to the organizations, and finally negotiate agreements that will meet the needs of both partners in the arrangement. This is a long-term process that requires a high degree of interaction and maintenance through time.

In the product/service sales model there is a requirement for marketing and promotion of the service as visibility and awareness of the service is a prerequisite for its use. Once awareness is established sufficient documentation and interface capabilities about the products and services of the platform must be produced to both demonstrate the capabilities and value of the system while also lowering the barriers to use.

Users – Market development is a core component of both the corporate support and product/service sales models. Without a clear understanding of the target market, potential corporate partners/sponsors can't be identified. Once the market is identified the relationship building, promotion and documentation activities described above must be used to actively develop and maintain connections with the identified market of users.

Communities – As highlighted above, engagement with the community of potential users is critical to the success of both the corporate and product/service sales models. As the funding generated by both of these models requires demonstration of community engagement (including use) or actual use lack of engagement puts both models at serious risk of failure.

Human Resources – The revenue generated through corporate support would typically consist of negotiated funding amounts that are delivered over a specified period of time, in many cases providing an opportunity for renewal if the continuation of the agreement appears beneficial to both parties. In the case of corporate sponsorship the agreement may be for a fixed amount while an agreement that provides in-kind contribution of resources may provide resources in proportion to needs up to specified cap. In the case of a corporate membership model the funding provided may potentially fall into different membership levels, with the membership tiers providing differential member benefits to the member companies. The indirect membership model may provide either fixed funding or proportional funding based on the number of organization members that will have access to the software.

The product/service sales model provides a revenue stream that is directly proportional to the number of items "sold" or delivered. In this case the actual revenue generated may scale with demand on the system, but may also fall below needed minimum levels for maintenance of core system functionality (i.e. base operating costs).

Software Engineering – In the case of both corporate support and product/service sales the system must have capabilities that are well aligned with the needs and requirements of the sponsoring organizations or customers. In both cases the development of an agile development model that provides for an ongoing exchange between users/stakeholders and system developers must be developed. In such an agile development model emerging needs can be rapidly developed and tested and if proven beneficial can be integrated into the system with a minimum of time or effort.

Product management – The product management model must align with an overall platform that is flexible and extensible. This is required to meet the (often rapidly) evolving needs of the users and partners from whom funding is directly obtained. Without a responsive product management model the alignment of system capabilities with user/sponsor needs can drift apart, yielding an opportunity for rapid drop-off in obtained funding.

Advantages

- Model has the capability to provide steady, long-term funding
- This model can contribute to a business diversification strategy for resilience
- Could build partnerships with membership organizations (i.e. corporate [non-profit or otherwise] sponsors) to provide premium level service access

Challenges

- May require significant promotion and effective communication with corporate sponsors
- Expectations of "free" on the internet works against buy-in for the premium tier of freemium
- Tension between profit motive for Corporate partners and productive revenue capacity
- Challenge of differentiation between free service level and premium service level
- Association with some corporate sponsors may disenfranchise some users
- Providers of free data will not want distributors to profit from it if there is no added value

Consortium Model

Model Description

Organizations (Universities, non-profits, for-profits, foundations, individuals, etc.) establish a consortium based on a common mission to create efficient and enhanced use of resources. In this case, the Consortium provides a sustaining environment through provision of infrastructure, financial support, maintenance and community engagement. The Consortium can be a single infrastructure or may provide distributed support through its members.

Mapping Revenue Models onto Sustainment Attributes

Funding – Start up funding for the Consortium is often provided by a single entity (Foundation or other funding source). Continued support comes from Consortium membership fees (annual or other). Sustaining funding requires sustained membership by a critical mass of organizations or external funding sources may be required due to membership instability.

Users – Significant ongoing effort may be required to identify and maintain the user community beyond the Consortium members. Alternatively, the Consortium may represent the primary user communities and further marketing efforts may not be required.

Communities – Community engagement is a significant effort in the consortium model and may require full time staff to maintain and increase membership.

Human Resources – The Consortium is typically governed by a Board of Directors who provide oversight and long range planning. Software and systems development and maintenance is planned, budgeted and implemented on an annual basis to meet the prioritized needs and objectives of the Consortium. The support infrastructure may be distributed across member organizations, or can be centralized in a single facility. 25

Software Engineering – Software development and/or maintenance is provided internally by an expert/advocate within the consortium body or may be subcontracted. Software may also be developed externally and adopted by the Consortium. There is also the possibility of Consortium members contributing to the development and maintenance as well as infrastructure support.

Product management – The software is managed to meet the needs of the Consortium and the communities it serves. Input to this process comes through the Consortium members.